

Отчет по практической работе №4

Визуализация данных средствами Matplotlib. Основы

Цель работы: получить навыки использования библиотеки визуализации данных Matplotlib с использованием языка программирования Python.

Ход работы:

Заходим на сайт Google Colab и авторизуемся в свой Google аккаунт. Создаем новый файл и проверяем работу программы.

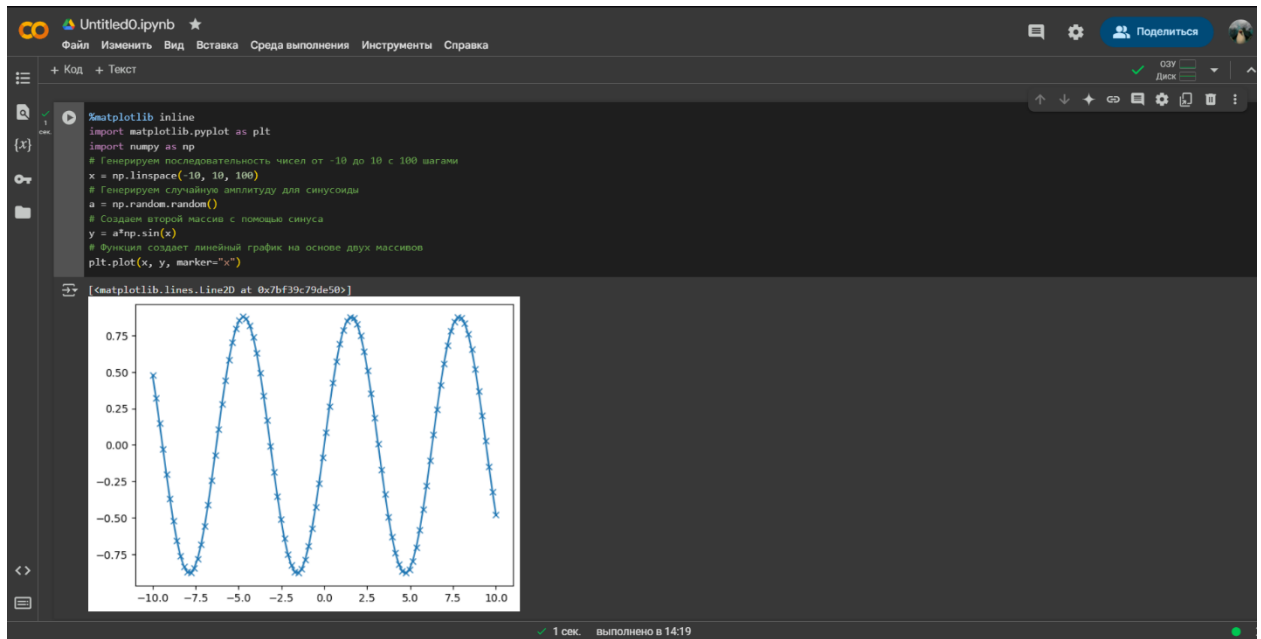


Рис 4.1

Импортируем необходимые библиотеки и загружаем данные из сети в pandas dataframe. Выводим первые 5 элементов.

```
[2] import numpy as np
import pandas as pd

[3] df_can = pd.read_excel('https://s3-api.us-geo.objectstorage.softlayer.net/cf-courses-data/CognitiveClass/DV0101EN/labs/Data_Files/Canada.xlsx',
    sheet_name='Canada by Citizenship',
    skiprows=range(20),
    skipfooter=2
)
print('Данные загружены и записаны в dataframe!')
```

Данные загружены и записаны в dataframe!

df_can.head()

	Type	Coverage	OdName	AREA	AreaName	REG	RegName	DEV	DevName	1980	...	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013
0	Immigrants	Foreigners	Afghanistan	935	Asia	5501	Southern Asia	902	Developing regions	16	...	2978	3436	3009	2652	2111	1746	1758	2203	2635	2004
1	Immigrants	Foreigners	Albania	908	Europe	925	Southern Europe	901	Developed regions	1	...	1450	1223	856	702	560	716	561	539	620	603
2	Immigrants	Foreigners	Algeria	903	Africa	912	Northern Africa	902	Developing regions	80	...	3616	3626	4807	3623	4005	5393	4752	4325	3774	4331
3	Immigrants	Foreigners	American Samoa	909	Oceania	957	Polynesia	902	Developing regions	0	...	0	0	1	0	0	0	0	0	0	0
4	Immigrants	Foreigners	Andorra	908	Europe	925	Southern Europe	901	Developed regions	0	...	0	0	1	1	0	0	0	0	1	1

5 rows x 43 columns

Рис 4.2

Удаляем неинформативные столбцы и повторно выводим первые 5 элементов.

```
[5] print(df_can.shape)
(195, 43)

df_can.drop(['AREA', 'REG', 'DEV', 'Type', 'Coverage'], axis=1, inplace=True)
df_can.head()
```

	OdName	AreaName	RegName	DevName	1980	1981	1982	1983	1984	1985	...	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013
0	Afghanistan	Asia	Southern Asia	Developing regions	16	39	39	47	71	340	...	2978	3436	3009	2652	2111	1746	1758	2203	2635	2004
1	Albania	Europe	Southern Europe	Developed regions	1	0	0	0	0	0	...	1450	1223	856	702	560	716	561	539	620	603
2	Algeria	Africa	Northern Africa	Developing regions	80	67	71	69	63	44	...	3616	3626	4807	3623	4005	5393	4752	4325	3774	4331
3	American Samoa	Oceania	Polynesia	Developing regions	0	1	0	0	0	0	...	0	0	1	0	0	0	0	0	0	0
4	Andorra	Europe	Southern Europe	Developed regions	0	0	0	0	0	0	...	0	0	1	1	0	0	0	0	1	1

5 rows x 38 columns

Рис 4.3

Приводим данные к более удобному виду и выводим первые 5 элементов. Проверяем, являются ли наименования всех столбцов строками.

```
df_can.rename(columns={'OdName': 'Country', 'AreaName': 'Continent', 'RegName': 'Region'}, inplace=True)
df_can.head()

all(isinstance(column, str) for column in df_can.columns)
```

	Country	Continent	Region	DevName	1980	1981	1982	1983	1984	1985	...	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013
0	Afghanistan	Asia	Southern Asia	Developing regions	16	39	39	47	71	340	...	2978	3436	3009	2652	2111	1746	1758	2203	2635	2004
1	Albania	Europe	Southern Europe	Developed regions	1	0	0	0	0	0	...	1450	1223	856	702	560	716	561	539	620	603
2	Algeria	Africa	Northern Africa	Developing regions	80	67	71	69	63	44	...	3616	3626	4807	3623	4005	5393	4752	4325	3774	4331
3	American Samoa	Oceania	Polynesia	Developing regions	0	1	0	0	0	0	...	0	0	1	0	0	0	0	0	0	0
4	Andorra	Europe	Southern Europe	Developed regions	0	0	0	0	0	0	...	0	0	1	1	0	0	0	0	1	1

5 rows x 38 columns

False

Рис 4.4

Изменяем наименование всех столбцов, чтоб они имели тип string. Приводим данные к более удобному виду и выводим первые 5 элементов.

```
df_can.columns = list(map(str, df_can.columns))
all(isinstance(column, str) for column in df_can.columns)

df_can.set_index('Country', inplace=True)
df_can.head()
```

	Continent	Region	DevName	1980	1981	1982	1983	1984	1985	1986	...	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013
Afghanistan	Asia	Southern Asia	Developing regions	16	39	39	47	71	340	496	...	2978	3436	3009	2652	2111	1746	1758	2203	2635	2004
Albania	Europe	Southern Europe	Developed regions	1	0	0	0	0	0	1	...	1450	1223	856	702	560	716	561	539	620	603
Algeria	Africa	Northern Africa	Developing regions	80	67	71	69	63	44	69	...	3616	3626	4807	3623	4005	5393	4752	4325	3774	4331
American Samoa	Oceania	Polynesia	Developing regions	0	1	0	0	0	0	0	...	0	0	1	0	0	0	0	0	0	0
Andorra	Europe	Southern Europe	Developed regions	0	0	0	0	0	0	2	...	0	0	1	1	0	0	0	0	1	1

5 rows x 37 columns

Рис 4.5

Добавляем новый столбец Total, который будет являться суммой всех столбцов. Создаем новый набор данных на базе предыдущего и выделяем в него 5 стран, иммиграция из которых больше всех остальных.



Рис 4.6

Выводим данные в виде графика типа «Диаграмма с областями».

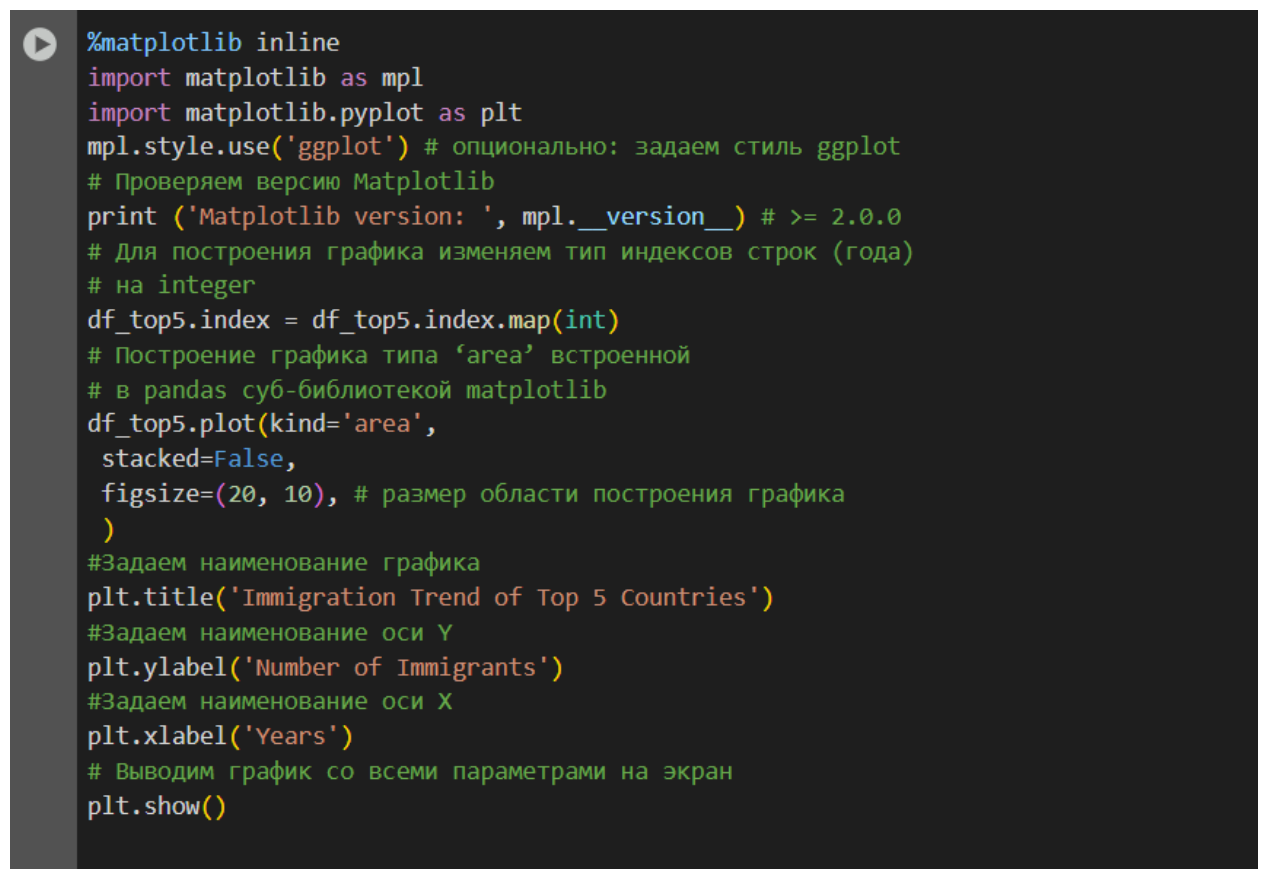


Рис 4.7

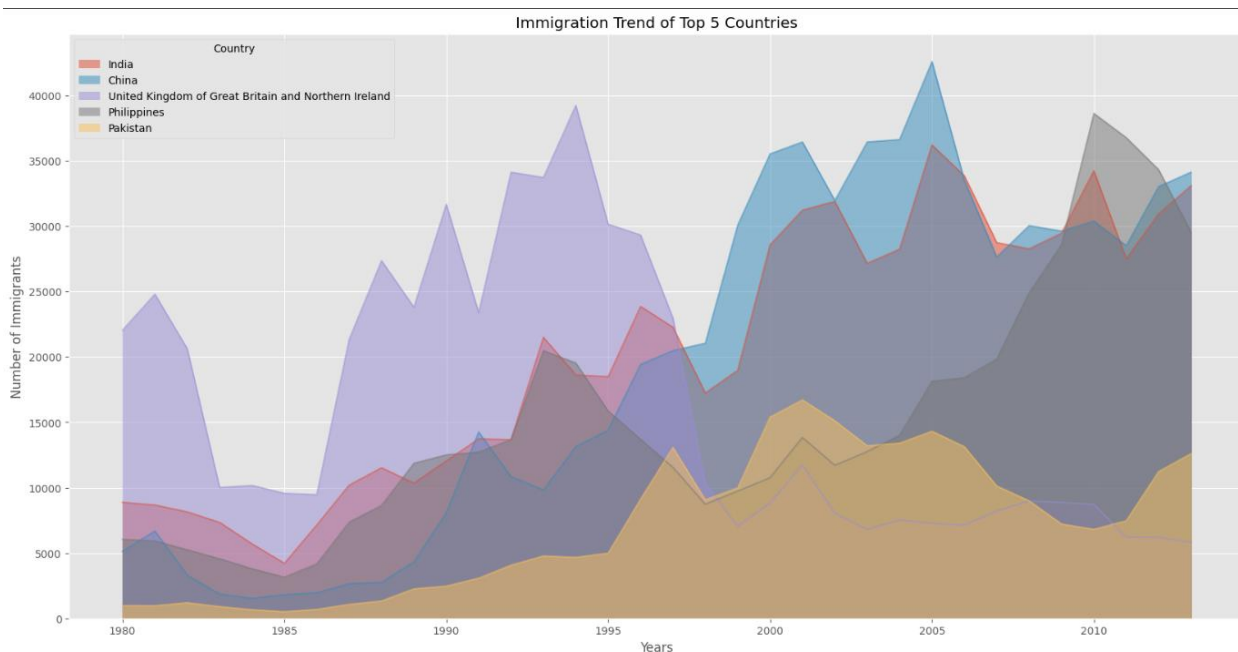


Рис 4.8

Вывод: в результате выполнения практической работы были приобретены навыки использования библиотеки визуализации данных Matplotlib с использованием языка программирования Python. Был создан 2D график без загрузки дополнительных данных. Для 2 графика были загружены данные по иммиграции из различных стран. Были удалены ненужные столбцы и проведено форматирование данных в более удобный вид. Построена диаграмма по данным 5 стран с наибольшей иммиграцией.